

# APLIKASI PENGENALAN DAUN UNTUK KLASIFIKASI TANAMAN DENGAN METODE *PROBABILISTIC NEURAL NETWORK*

<sup>1</sup> Gregorius Satia Budhi

<sup>2</sup> Tok Fenny Handayani

<sup>3</sup> Rudy Adipranata

<sup>1, 2, 3</sup> Teknik Informatika Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya

<sup>1</sup>email : greg@petra.ac.id

## ABSTRAK

*Dalam kehidupan sehari-hari, manusia sering melihat tanaman di sekitarnya dengan ciri-ciri yang beranekaragam. Keanekaragaman tanaman tampak pada bentuk daunnya seperti oval, waru (cordate), elips dan lain-lain. Akan tetapi, pada kenyataan masih banyak orang belum mengenal nama-nama tanaman karena banyaknya keanekaragaman jenis tanaman tersebut. Seiring dengan berkembangnya kemajuan di bidang teknologi informasi, maka dimungkinkan untuk mengembangkan aplikasi guna melakukan pengenalan daun yang dapat membantu orang dalam mengenali jenis tanaman tersebut. Pada penelitian ini dikembangkan aplikasi dengan menggunakan metode Probabilistic Neural Network untuk melakukan pengenalan daun. Sebagai input data digunakan gambar daun dan dilakukan feature extraction untuk mendapatkan ciri-ciri yang mewakili bentuk daun. Dari hasil pengujian, tingkat keakuratan pengenalan mencapai sekitar 70% dengan menggunakan spread 0.25 hingga 0.5.*

*Kata Kunci: probabilistic neural network, pengenalan daun, klasifikasi tanaman*

## 1. PENDAHULUAN

Di dunia ini terdapat beraneka ragam jenis tanaman yang memiliki ciri-ciri yang berbeda antara tanaman yang satu dengan lainnya. Karena begitu beragamnya tanaman tersebut maka menimbulkan gagasan untuk menciptakan suatu sistem guna mengenal dan mempelajarinya. Beberapa ahli Biologi mencoba menciptakan suatu sistem untuk mempermudah mengenal dan mempelajari organisme melalui suatu cara pembuatan klasifikasi. Pengklasifikasian merupakan proses pengelompokan berdasarkan ciri-ciri tertentu. Salah satu bagian yang dapat digunakan untuk pengklasifikasian tanaman adalah bentuk daun dari tanaman tersebut.

Tetapi hingga saat ini ternyata tidak banyak orang yang mengetahui nama-nama tanaman karena begitu banyaknya keanekaragaman tanaman tersebut, sehingga dibutuhkan cara mudah untuk dapat mengenali tanaman tersebut. Seiring

dengan kemajuan teknologi informasi, dimungkinkan untuk mengembangkan aplikasi guna pengenalan daun. Untuk itu pada penelitian ini dikembangkan aplikasi untuk mengenali tanaman melalui daun dengan menggunakan metode *Probabilistic Neural Network*.

## 2. TINJAUAN PUSTAKA

*Probabilistic Neural Network* (PNN) dikembangkan pertama kali oleh Donald Specht. *Probabilistic Neural Network* adalah suatu metode jaringan saraf tiruan (*neural network*) yang menggunakan pelatihan (*training*) *supervised*. PNN biasanya digunakan untuk masalah klasifikasi. Secara garis besar, PNN mempunyai tiga *layer* yaitu (Haykin, 1998):

- *input layer*

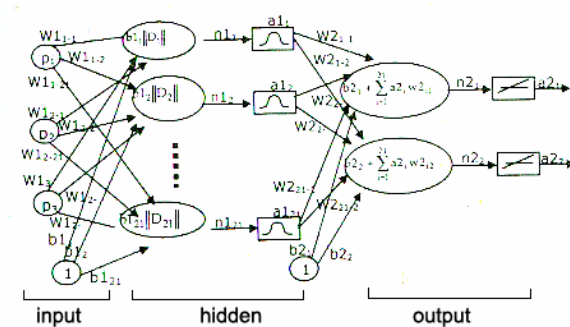
*Input layer* merupakan *layer* data input bagi PNN.

- *hidden layer*

Pada layer ini menerima data dari *input layer* yang akan diproses dalam PNN.

- *output layer*

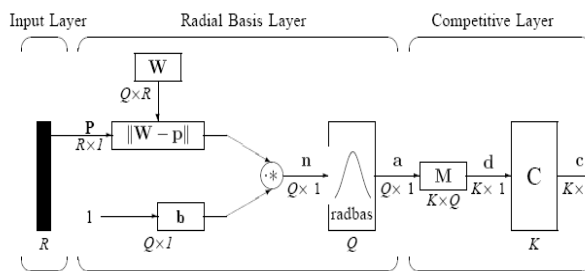
Pada layer ini, *node output* berupa *binary* yang menghasilkan keputusan klasifikasi.



Gambar 1. Probabilistic Neural Network

### Struktur Network PNN

Struktur PNN lebih detail yang digunakan dalam penelitian ini terdiri dari 3 bagian yaitu *input layer*, *radial basis layer* dan *competitive layer* (Wu, 2007). *Radial Basis Layer* melakukan evaluasi jarak vektor antara vektor *input* dan baris *weight* dalam matriks bobot (*weight*). Jarak tersebut diskala oleh *radial basis function nonlineary*. Pada *competitive layer* dilakukan pencarian jarak terpendek dan menemukan *training pattern* dari *input pattern* berdasarkan jaraknya. Berikut ini adalah struktur PNN lebih detail yang diilustrasikan pada skema dibawah ini :



Gambar 2. Struktur PNN

Pada *input layer* terdapat vektor *input* ditunjukkan dengan P, digambarkan sebagai vertikal hitam pada Gambar 2 dengan dimensi R x 1. *Radial Basis Layer* (RBF) merupakan *hidden unit* dalam PNN. Dalam *radial basis layer*, jarak vektor antara vektor *input* P dan vektor *weight*

(W) dibuat dari tiap baris dari matriks bobot / *weight* (W) yang dikalkulasi. Jarak antara vektor didefinisikan sebagai *dot product* antara dua vektor. Diasumsikan dimensi W adalah Q x R. *Dot product* antara p dan baris ke-i dari yang dihasilkan W elemen ke-i dari jarak vektor  $\|W - P\|$ , dimana dimensinya adalah Q x 1, yang ditunjukkan pada gambar diatas. Simbol “-” menyatakan jarak antar vektor. Kemudian, vektor *bias* b dikombinasi dengan  $\|W - P\|$  dengan multiplikasi *element-by-element*, yang digambarkan dengan “.” pada Gambar 2. Hasilnya ditunjukkan sebagai :

$$n = \|W - P\| .* b \quad (1)$$

dimana  $b = \sqrt{-\ln 0.5} / s$ , s adalah konstan *spread* dari PNN.

Pada fungsi transfer dalam PNN membuat kriteria jarak dengan memberi respon ke tengah, yang didefinisikan sebagai :

$$radbas(n) = e^{-n^2} \quad (2)$$

Tiap elemen dari n di substitusi dan menghasilkan elemen a, vektor *output* dari *radial basis layer*, digambarkan elemen ke-i dari a sebagai :

$$a_i = radbas (\|W_i - P\| .* b_i) \quad (3)$$

dimana  $W_i$  adalah vektor yang dibuat dari baris ke-i dari W dan  $b_i$  adalah elemen ke-i dari vektor bias b.

Pada *competitive layer*, vektor a dikalikan dengan matriks *layer weight* M ( $W_2$ ), menghasilkan vektor *output* d.

$$a_{11} * W_{2K1} + a_{12} * W_{2K2} + \dots + a_{1R} * W_{2KR} + b = T_{K1} \quad (4)$$

Fungsi *competitive* digambarkan dengan C, menghasilkan 1 sebagai elemen yang terbesar dari d dan 0 sebagai elemen terkecil dari d. Vektor keluaran dari *competitive function* digambarkan sebagai c. Indeks 1 dapat digunakan sebagai indeks untuk mencari nama latin dari tanaman.

### Algoritma PNN

Langkah-langkah pelatihan pada PNN adalah sebagai berikut :

- o Inisialisasi

- Melakukan inisialisasi bobot (*weight*) awal pada *radial basis layer* yang dilambangkan sebagai *W* dengan *transpose* matriks *R x Q* dari vektor *training*. Tiap baris dari *W* terdiri dari 12 morfologi digital daun dari satu contoh yang di *training*.
- Melakukan inisialisasi bobot bias.
  - Menghitung jarak (*distance*) dari data *input* (*P*) dengan bobot awal (*W*).
  - Menghitung nilai aktivasi dari jarak antara bobot awal dengan data *input* (*W-P*), dengan menggunakan fungsi radial basis (*radbas*).
  - Mencari bobot baru dan bobot bias yang baru dengan menggunakan metode LMS
  - Masuk ke dalam *competitive layer*, menghitung *output* dari jaringan.
  - Simpan bobot awal dan bobot akhir ke dalam *database* berupa *file*.

Sedangkan untuk langkah-langkah *recognition* (pengenalan) adalah sebagai berikut :

- Mengambil target, bobot awal dan bobot akhir dari *database*
- Meneruskan sinyal *input* ke *hidden layer* (*radial basis layer*) dan mencari jarak antara data *input* dengan bobot awal.
- Mencari nilai aktivasi dengan menggunakan fungsi radial basis (*radbas*).
- Menghitung *output* dari jaringan
- Membandingkan hasil *output* dengan target.
- Menampilkan hasil pengenalan.

### Feature Extraction

Feature extraction yang dilakukan terdapat 17 jenis yang terdiri dari 5 fitur geometri dasar serta 12 fitur morfologi digital. Lima fitur geometri dasar adalah diameter, panjang, lebar, luar serta perimeter daun (Parker, 1994), sedangkan 12 fitur morfologi digital adalah *smooth factor*, *aspect ratio*, *form factor*, *rectangularity*, *narrow factor*, rasio perimeter dari diameter, rasio perimeter

dengan panjang dan lebar daun serta 5 macam vein feature (Wu, 2007). Data yang akan digunakan sebagai input pada PNN adalah 12 fitur morfologi digital tersebut.

*Smooth factor* didefinisikan sebagai rasio antara luas dari gambar daun yang dilakukan *smoothing* dengan *5\*5 rectangular averaging filter* dan *2\*2 rectangular averaging filter*. Aspect Ratio didefinisikan sebagai rasio dari panjang daun dengan lebar daun.

$$AspectRatio = \frac{Lp}{Wp} \quad (5)$$

*Lp* = panjang daun  
*Wp* = lebar daun

*Form factor* menggambarkan perbedaan antara sebuah daun dan lingkaran

$$FF = \frac{4\pi * A}{P^2} \quad (6)$$

*A* = luas daun  
*P* = keliling daun

*Rectangularity* merupakan kemiripan antara daun dan *rectangularity*.

$$rect = \frac{Lp * Wp}{A} \quad (7)$$

*A* = luas daun  
*Wp* = lebar daun  
*Lp* = panjang daun

*Narrow factor* menjelaskan rasio dari diameter dengan panjang daun.

$$NarrowFactor = \frac{D}{Lp} \quad (8)$$

*D* = diameter daun  
*Lp* = panjang daun

Rasio perimeter dari diameter dinyatakan pada rumus (9).

$$P/D \quad (9)$$

*D* = diameter daun  
*P* = keliling daun

Rasio perimeter dengan panjang dan lebar daun dinyatakan pada rumus (10).

$$\frac{P}{(Lp + Wp)} \quad (10)$$

*P* = keliling daun  
*Wp* = keliling daun  
*Lp* = panjang daun

*Vein Feature* didapatkan dengan menggunakan *morphology opening* pada

gambar *grayscale* dengan *flat*, *disk-shapped* yang menyusun elemen dengan radius 1,2,3,4 dan mengurangi dengan *margin* daun. Hasilnya adalah bentuk gambar seperti *vein* (urat daun).

$$\text{Feature 1 : } A_{v1} / A \quad (11)$$

$$\text{Feature 2 : } A_{v2} / A \quad (12)$$

$$\text{Feature 3 : } A_{v3} / A \quad (13)$$

$$\text{Feature 4 : } A_{v4} / A \quad (14)$$

$$\text{Feature 5 : } A_{v4} / A_{v1} \quad (15)$$

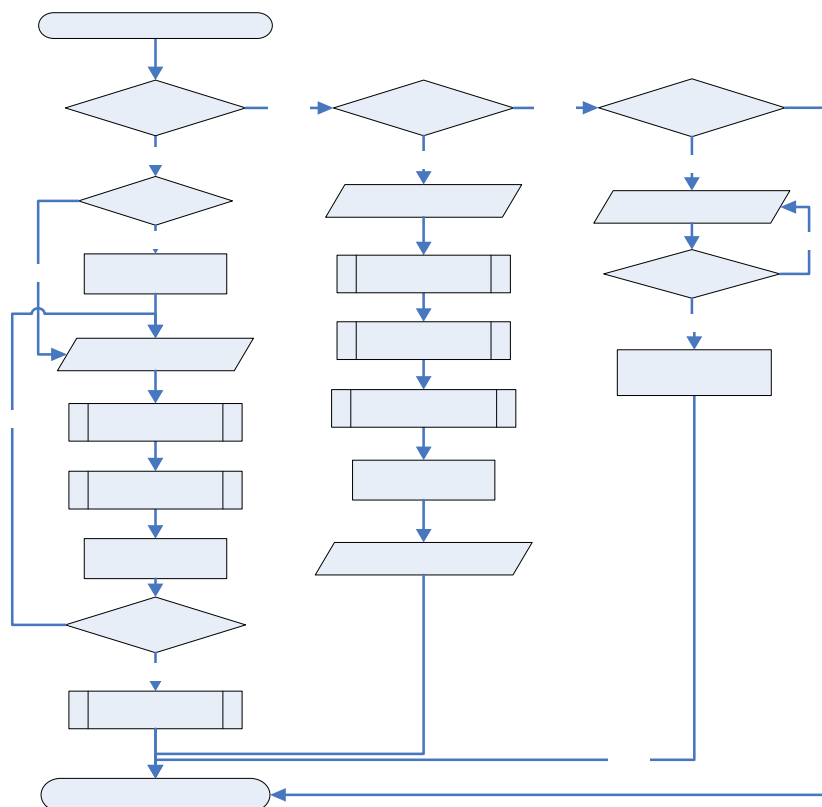
### 3. METODE PENELITIAN

Pada penelitian ini dilakukan empat hal utama yaitu pengambilan sampling guna *input* berupa gambar-gambar daun, pemrosesan citra, *feature extraction*, serta pelatihan PNN. Jenis daun yang akan dilatihkan ke sistem berjumlah 20 jenis dan pada masing-masing jenis dilakukan pelatihan sebanyak 10, 20 serta 30 daun. Untuk mendapatkan citra digital dari

masing-masing daun, dilakukan *scanning* terhadap daun yang didapat. Dari hasil *scanning* tersebut dilakukan pemrosesan citra digital terlebih dahulu sebelum dilakukan pengambilan *feature extraction* yaitu 12 morfologi digital. Hasil dari *feature extraction* selanjutnya digunakan sebagai *input* bagi PNN untuk dilakukan pelatihan. Setelah proses pelatihan selesai, dilakukan pengujian dengan menggunakan data sampling yang berbeda pada masing-masing jenis untuk mengetahui keakuratan pengenalan sistem.

### 4. HASIL DAN PEMBAHASAN

Untuk memberikan gambaran sistem keseluruhan, ditampilkan desain dengan menggunakan diagram alir seperti terlihat pada Gambar 3.



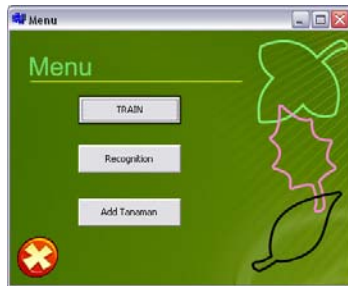
Gambar 3. Diagram Alir Sistem

Pada gambar *input*, terlebih dahulu dilakukan pemrosesan gambar, yaitu dilakukan perubahan dari gambar berwarna menjadi *grayscale*, kemudian dilakukan

*thresholding* untuk mendapatkan bentuk biner dari gambar. Setelah itu dilakukan *smoothing* serta *boundary enhancement* dengan menggunakan *Laplacian filter*

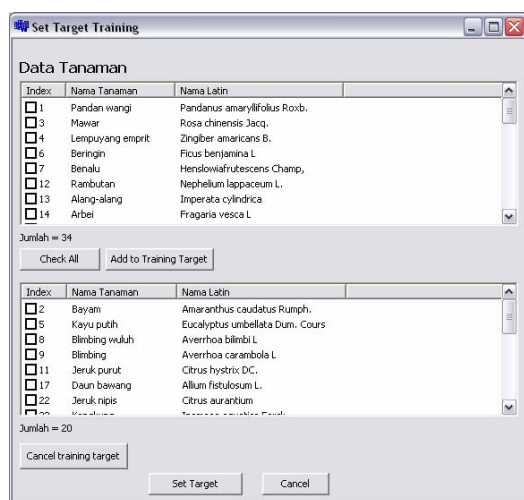
(Gonzales, 2002). Setelah pemrosesan gambar selesai dilakukan, barulah dilakukan *feature extraction* serta pelatihan ataupun pengenalan dengan menggunakan PNN.

Menu utama hasil aplikasi yang dikembangkan terdapat pada Gambar 4.



Gambar 4. Hasil Aplikasi

Dari menu tersebut terdapat tiga hal yang dapat dilakukan yaitu pelatihan, pengenalan serta *update* data tanaman yang berada di *database*. Data tanaman yaitu nama terdapat dalam database dimana data tersebut dapat dipilih mana yang akan dilakukan pelatihan. Hal ini dapat dilihat pada Gambar 5.

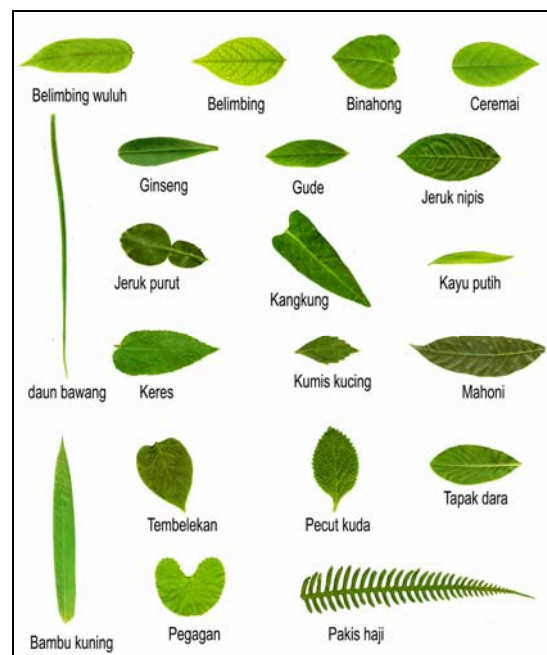


Gambar 5. Pemilihan Data yang Dilakukan Pelatihan

Adapun data yang diuji cobakan pada sistem adalah 20 jenis daun seperti terdapat pada Tabel 1 dengan gambar seperti terdapat pada Gambar 6.

Tabel 1. Jenis Tanaman Uji Coba

| No | Tanaman Target |
|----|----------------|
| 1  | Blimbing wuluh |
| 2  | Blimbing       |
| 3  | Daun bawang    |
| 4  | Jeruk nipis    |
| 5  | Kangkung       |
| 6  | Pakis          |
| 7  | Kumis kucing   |
| 8  | Mahoni         |
| 9  | Pegagan        |
| 10 | Bambu kuning   |
| 11 | Jeruk purut    |
| 12 | Tembelean      |
| 13 | Ceremai        |
| 14 | Gude           |
| 15 | Pecut Kuda     |
| 16 | Tapak dara     |
| 17 | Keres          |
| 18 | Kayu putih     |
| 19 | Ginseng        |
| 20 | Binahong       |



Gambar 6. Gambar Daun Ujicoba

Pengujian dilakukan dengan menggunakan 20 jenis spesies daun dimana jumlah masing-masing spesies yang dilakukan pelatihan berbeda yaitu 10, 20 serta 30. Selain itu juga dilakukan variasi untuk menggunakan beberapa nilai *spread* dengan *range* antara 0.01 hingga 1. Hasil dari pengenalan ada pada Tabel 2.

Tabel 2. Hasil Pengenalan

| Jumlah spesies | Jumlah data training per spesies | Spread | Accuracy | Precision | Error rate |
|----------------|----------------------------------|--------|----------|-----------|------------|
| 20             | 10                               | 0.01   | 10.409%  | 92.466%   | 89.591%    |
|                |                                  | 0.05   | 39.398%  | 61.398%   | 60.602%    |
|                |                                  | 0.1    | 48.432%  | 62.932%   | 51.568%    |
|                |                                  | 0.25   | 62.136%  | 68.136%   | 37.864%    |
|                |                                  | 0.5    | 69.466%  | 72.466%   | 30.534%    |
|                |                                  | 0.75   | 70.932%  | 73.432%   | 29.068%    |
|                |                                  | 1      | 68.182%  | 78.182%   | 31.818%    |
|                | 20                               | 0.01   | 10.909%  | 89.341%   | 89.091%    |
|                |                                  | 0.05   | 52.682%  | 57.807%   | 47.318%    |
|                |                                  | 0.1    | 54.932%  | 65.432%   | 45.068%    |
|                |                                  | 0.25   | 71.602%  | 73.102%   | 28.398%    |
|                |                                  | 0.5    | 68.398%  | 69.898%   | 31.602%    |
|                |                                  | 0.75   | 62.068%  | 67.568%   | 37.932%    |
|                |                                  | 1      | 58.273%  | 68.773%   | 41.727%    |
|                | 30                               | 0.01   | 15.534%  | 86.841%   | 84.466%    |
|                |                                  | 0.05   | 60.886%  | 65.886%   | 39.114%    |
|                |                                  | 0.1    | 68.511%  | 73.011%   | 31.489%    |
|                |                                  | 0.25   | 74.602%  | 76.102%   | 25.398%    |
|                |                                  | 0.5    | 73.568%  | 75.068%   | 26.432%    |
|                |                                  | 0.75   | 64.443%  | 64.443%   | 35.557%    |
|                |                                  | 1      | 58.273%  | 68.773%   | 41.727%    |

Dari hasil pengujian tersebut terlihat bahwa *error rate* tinggi terjadi pada saat penggunaan nilai *spread* 0.01. dan nilai *error rate* terkecil terjadi pada penggunaan *spread* 0.25 hingga 0.5.

## 5. KESIMPULAN DAN SARAN

- Berdasarkan pengujian data, nilai *spread* yang baik adalah antara 0.25 hingga 0.5 karena mempunyai nilai *error-rate* terendah setelah diujikan untuk 20 jenis (spesies) daun.
- Jumlah minimal sampel yang dilatihkan per jenis adalah sebanyak jumlah jenis (spesies) yang dilatihkan. Semakin banyak jumlah sampel yang dilatihkan per jenis daun, maka semakin akurat atau semakin baik hasil yang didapatkan. Dan sebaliknya, semakin sedikit jumlah sampel yang dilatihkan, maka hasil yang didapatkan semakin tidak akurat.
- Kesalahan deteksi terjadi karena daun tersebut mempunyai ciri-ciri morfologi yang sama ataupun mirip, seperti bentuk daun dan urat daun. Jadi pada waktu pelatihan dan pengenalan lebih

baik menggunakan daun yang sudah dewasa, dikarenakan bentuk dan ciri dari daun tersebut tidak akan berubah.

## DAFTAR PUSTAKA

- [1] Gonzales, Rafael C. & Woods, Richard E. 2002. *Digital image processing (2<sup>nd</sup> ed.)*. Prentice Hall, New Jersey.
- [2] Haykin, Simon. 1998. *Neural networks a comprehensive foundation (2<sup>nd</sup> ed.)*. Prentice Hall International, Inc, Hamilton.
- [3] Parker, J.R. 1994. *Practical computer vision using C*. John Wiley & Sons, USA.
- [4] Wu, Stephen Gang. 2007. A leaf recognition algorithm for plant clasification using probabilistic neural network. *IEEE ISSPIT 2007 on Computer Science and Electrical Engineering involve Artificial Intelligence and Neurology*.